RELIABLE NUMERICAL COMPUTATION IN AN OPTIMAL OUTPUT-FEEDBACK DESIGN

Brett VanSteenwyk and Uy-Loi Ly *
Department of Aeronautics and Astronautics, FS-10
University of Washington
Seattle, WA 98195

May 8, 1991

Abstract

This paper presents a reliable algorithm for the evaluation of a quadratic performance index and its gradients with respect to the controller design parameters. The algorithm is part of a design algorithm for optimal linear dynamic output-feedback controller that minimizes a finite-time quadratic performance index. The numerical scheme is particularly robust when it is applied to the control-law synthesis for systems with densely packed modes and where there is a high likelihood of encountering degeneracies in the closed-loop eigensystem. This approach through the use of an accurate Padé series approximation does not require the closed-loop system matrix to be diagonalizable. The algorithm has been included in a control design package for optimal robust low-order controllers. Usefulness of the proposed numerical algorithm has been demonstrated using numerous practical design cases where degeneracies occur frequently in the closed-loop system under an arbitrary controller design initialization and during the numerical search.

1 Introduction

Traditional design methods in linear optimal control for continuous-time systems have been extensively treated in recent literature [1]. Development of these control systems are usually based on characterization of the control problem under the setting of optimization of the two-norm of a set of controlled output responses to random disturbance inputs or initial conditions. Additional consideration of design robustness is taken by formulating the problem to include H^∞-norm bound constraints for a class of additive and multiplicative uncertainties applied at the plant inputs and/or outputs. Solutions are obtained for both the state- and output-feedback design problems and involve in the majority of cases solving an appropriate set of algebraic Riccati equations [2, 3]. Theoretical studies of these approaches have been the major concern of researchers in the control field and major breakthrough has been made in recent work by Stoorvogel [4, 5]. An alternate and seldomly mentioned design option for robust multivariable control of linear time-invariant systems is based on direct numerical optimization of a quadratic performance index with an arbitrarily specified controller structure. We believe that careful formulation of the design problem under nonlinear constrained optimization can be of great value in the synthesis of robust multivariable control systems.

Early work in this area have been published by Levine and Athans [6], Anderson and Moore [7], and extensive review of the subject was performed by Makila [8]. Recently, a new look into parameter optimization to multivariable control synthesis is provided by Ly [9] where he used a quadratic performance based on finite-time horizon. In the latter work, a numerical optimization technique based on [10] was used. At each design iteration it requires a precise evaluation of a finite-time quadratic performance index and its gradients with respect to the design parameters. Analytical expressions have been developed to evaluate these quantities under the key assumption that the closed-loop system being diagonalizable. This assumption is found to be unsatisfactory and is the cause of convergence difficulties in the iterative search when it attempts to invert an ill-conditionned eigenvector matrix for the diagonalization. The work presented in this paper is to resolve this numerical difficulty and thereby extends the results of Ly [9] for cases where the closed-loop systems are degenerate, i.e the closed-loop system has

*The work of B. VanSteenwyk and U. Ly is supported in part by NASA Ames Research Center under grant contract NAG-2-691.

repeated eigenvalues and the corresponding set of system eigenvectors does not span the whole state-space of the closed-loop system.

The paper is organized as follows. Section 2 reviews the problem formulation for a linear optimal control design using direct parameter optimization. Analytical expressions for the evaluation of the quadratic cost function and its gradients with respect to the controller design parameters are also given in section 2. The current approach to evaluate these quantities are briefly reviewed in section 3. An alternate numerical scheme for the exponential matrix and convolution integrals involving exponential matrices is presented in section 4. Approximation methods for the evaluation of these matrix quantities using Padé series are described in details in sections 5, 6, and 7. A design algorithm based on these numerical solution schemes has been incorporated into a computer-aided design package described in [9]. A simple design problem to motivate the need for a numerical algorithm that handles degeneracies in the closed-loop system matrix is given in section 8. Optimal solutions are obtained using the proposed method and the diagonalization method from [9]. The numerical algorithm has also been applied to the synthesis of an active control system for the JPL large space structure developed under the LSCL research program [16]. Results of this application are presented in section 9. Conclusions are given in section 10.

2 Problem Formulation

In this section, we recall the problem formulation described in Ly [9] for the control synthesis of a robust low-order controller in a linear time-invariant system. The system $P^i(s)$ is controlled by a constant-gain controller $C(s)$ as depicted in Figure 1 where $y_c^i(s)$ is the controlled output vector, $y_s^i(s)$ the measurement output vector, $w^i(s)$ the disturbance input vector and $u^i(s)$ the control input vector. As a consistent notation, the superscript i is used throughout this paper to denote the system model at the i^{th} plant condition. Note that the controller $C(s)$ is considered to be *fixed*, i.e does not vary with the design condition. It is modelled as a linear time-invariant system of arbitrary order whose formulation accomodates both a feedforward and a feedback controller structures. Robustness requirement in the context of our problem formulation is defined under the conditions that the control system $C(s)$ stabilizes the plant $P^i(s)$ over a class of design conditions ($i = 1, N_p$).

State equations describing the system model $P^i(s)$ of Figure 1 are as follows. Notice that, in the problem formulation, we assume without loss of generalities that all the system states are initially acquiescent. This assumption is not restrictive since one can always establish impulsive inputs $w^i(t)$ together with the appropriate influence matrix to represent any state initial conditions. At the i^{th} plant condition, the system design model is described by equations (1)-(3) below.

State Equations:

$$\begin{cases} \dot{x}^i(t) = F^i x^i(t) + G^i u^i(t) + \Gamma^i w^i(t) \\ x^i(0) = 0 \end{cases} \quad (1)$$

where $x^i(t)$ is a $n \times 1$ plant state vector, $u^i(t)$ an $m \times 1$ control vector, $w^i(t)$ an $m' \times 1$ disturbance-input vector, F^i an $n \times n$ state matrix, G^i an $n \times m$ control distribution matrix and Γ^i an $n \times m'$ input-disturbance distribution matrix.

Measurement Equations:

$$y_s^i(t) = H_s^i x^i(t) + D_{su}^i u^i(t) + D_{sw}^i w^i(t) \quad (2)$$

where $y_s^i(t)$ is a $p \times 1$ measurement vector, H_s^i a $p \times n$ state-to-measurement distribution matrix, D_{su}^i a $p \times m$ control-to-measurement distribution matrix and D_{sw}^i a $p \times m'$ input-disturbance-to-measurement distribution matrix.

Criterion Equations:

$$y_c^i(t) = H_c^i x^i(t) + D_{cu}^i u^i(t) + D_{cw}^i w^i(t) \quad (3)$$

where $y_c^i(t)$ is a $p' \times 1$ criterion vector, H_c^i a $p' \times n$ state-to-criterion distribution matrix, D_{cu}^i a $p' \times m$ control-to-criterion distribution matrix and D_{cw}^i a $p' \times m'$ input-disturbance-to-criterion distribution matrix.

For generality, the disturbances $w^i(t)$ are modeled as outputs of a linear time-invariant system excited by either impulse inputs or white noises. In this manner, one can shape the disturbance signals to have any deterministic responses (e.g filtered step functions, sinusoidal functions, exponentially decayed or growing sinusoidal functions, etc...) or to model stochastic inputs with any given power spectral density functions. At the i^{th} plant condition, the disturbance model is given by equations (4)-(5) below.

Disturbance State Equations:

$$\begin{cases} \dot{x}_w^i(t) = F_w^i x_w^i(t) + \Gamma_w^i \eta^i(t) \\ x_w^i(0) = 0 \end{cases} \quad (4)$$

where $x_w^i(t)$ is a $n' \times 1$ disturbance state vector, $\eta^i(t)$ a $m' \times 1$ vector of either parameterized random impulses (i.e $\eta^i(t) = \eta_o^i \delta(t)$ with $E[\eta_o^i] = 0$, and $E[\eta_o^i \eta_o^{iT}] = W_o$), or white-noise processes $\eta^i(t)$ with zero mean and covariance $E[\eta^i(t)\eta^{iT}(\tau)] = W_o \delta(t -$

r). The matrix W_o is an $m' \times m'$ diagonal positive semi-definite matrix, F_w^i an $n' \times n'$ state matrix of the disturbance model and Γ_w^i an $n' \times m'$ input-distribution matrix.

Disturbance Output Equations:

$$w^i(t) = H_w^i x_w^i(t) + D_w^i \eta^i(t) \tag{5}$$

where $w^i(t)$ is a $m' \times 1$ disturbance output vector, H_w^i an $m' \times n'$ disturbance output matrix and D_w^i an $m' \times m'$ direct feedthrough distribution matrix.

State model of the controller $C(s)$ in Figure 1 is that of a linear time-invariant system described by equations (6)-(7) below.

Controller State Equations:

$$\begin{cases} \dot{z}(t) = Az(t) + By_s^i(t) \\ z(0) = 0 \end{cases} \tag{6}$$

where $z(t)$ is a $r \times 1$ controller state vector, A a $r \times r$ state matrix of the controller and B a $r \times p$ measurement-input distribution matrix.

Control Equations:

$$u^i(t) = Cz(t) + Dy_s^i(t) \tag{7}$$

where $u^i(t)$ is an $m \times 1$ feedback control vector, C an $m \times r$ control-output distribution matrix and D an $m \times p$ direct feedthrough matrix.

For control-law synthesis, all the elements of the controller state matrices can be chosen as design parameters and some of them can be left fixed at pre-assigned values. In addition, if needed, linear and nonlinear equality or inequality constraints can be established among the selected design parameters in order to ensure a particular design structure. For convenience in the derivation of the performance index and its gradients with respect to the controller design parameters, we define a matrix C_o that groups all the controller state matrices (A, B, C, D) in one compact form as follows,

$$C_o = \begin{bmatrix} D & C \\ B & A \end{bmatrix}_{(m+r)\times(p+r)} \tag{8}$$

Thus, specification of the matrix C_o will completely define the controller state model. Obviously, for the case of a static output-feedback design (i.e the controller order $r = 0$), we simply have $C_o = D$. In Section 8, we will formulate a control design problem based on the minimization of a performance index using the controller $C(s)$ defined in equations (6)-(7).

To examine the entire class of H^2-norm based control problems and to handle the problem of sensitivity to plant modeling uncertainties, we define the objective function given in equations (9) and (10). This formulation turns out to be versatile and well-posed for the setting of a nonlinear constrained optimization problem. However, depending on the types of disturbance model, that is whether the disturbance outputs $w^i(t)$ are responses to impulse or white-noise inputs, different definitions for the objective function are needed. They are given below.

(a) For random impulse inputs $\eta^i(t)$:

$$\begin{aligned} J(t_f) \;=\; & \tfrac{1}{2} \sum_{i=1}^{N_p} \{ W_p^i \int_0^{t_f} e^{2\alpha^i t} \\ & E\left[\; y_c^{iT}(t) Q^i y_c^i(t) + u^{iT}(t) R^i u^i(t) \;\right] dt \} \end{aligned} \tag{9}$$

The expectation operator $E[-]$ is over the ensemble of the random variables η_o^i in the parameterized impulse inputs $\eta^i(t) = \eta_o^i \delta(t)$. Control design problems formulated with the above performance index $J(t_f)$ are often classified under the category of deterministic control. Under this category are, for example, the familiar control problems of command tracking control, disturbance rejection of unwanted but known external input signals, implicit and explicit model-following designs, H^2-control to initial conditions and H^∞-control to sinusoidal inputs.

(b) For random white-noise inputs $\eta^i(t)$:

$$J(t_f) = \frac{1}{2} \sum_{i=1}^{N_p} W_p^i E_{\alpha^i} \left[\begin{array}{l} y_c^{iT}(t_f) Q^i y_c^i(t_f) + \\ u^{iT}(t_f) R^i u^i(t_f) \end{array} \right] \tag{10}$$

The expectation operator $E_{\alpha^i}[-]$ is over the ensemble of the random processes defined in the input variables $\eta^i(t)$ for a closed-loop system destabilized by a factor α^i. The destabilization effectively adds a value α^i to the diagonal elements of the closed-loop system matrix. With the given performance index, one can address the entire class of H^2-norm based control design problems. For example, we can solve for the linear quadratic regulator design (LQ), linear quadratic gaussian (LQG) design, loop transfer recovery (LTR),closed-loop transfer recovery (CLTR), model reduction based on a minimization of H^2-norm of the error.

Note that the performance indices given in equations (9) and (10) are evaluated to a finite-time horizon t_f. The use of a finite time plays a significant role in the implementation of a reliable design algorithm for the optimum steady-state solution. It should be recognized that the objective function is well-defined regardless of whether the feedback control-law is stabilizing or not. Furthermore, a class of problems associated with command tracking of neutrally stable or unstable target responses (e.g step and ramp commands, sinusoidal trajectories) are only tractable under the setting of a finite-time objective function but

not in the confine of a steady-state objective function where $t_f \rightarrow \infty$. In practice, steady-state results, whenever possible, are usually achieved when the terminal time t_f is equal to five or six times the slowest time constant in the closed-loop system responses.

There are other unique features, besides the concept of design to a finite terminal time t_f, that we have incorporated into the design objective function of equations (9)-(10). First of all, this objective function is not the usual quadratic cost function defined in traditional linear optimal control problems. It is instead a weighted average of quadratic performance indices evaluated over the entire set of design conditions ($i = 1, N_p$). Different weights are assigned to each plant condition through the scalar variable W_p^i where $W_p^i \geq 0$. Of course, if $N_p = 1$, then we recover the usual quadratic cost function for a single *nominal* design condition. The time-weighted factor $e^{2\alpha' t}$ further allows us to impose directly a stability requirement for the closed-loop eigenvalues. Namely, when a steady-state design has been achieved and the optimum objective function is bounded, then closed-loop system eigenvalues for the controllable modes will have real parts less than $-\alpha^i$. Finally, the weighting matrices Q^i and R^i are symmetric and positive *semi*-definite matrices. Note that our solution approach to the minimization of the objective function $J(t_f)$ based on nonlinear optimization does not require the control weighting matrix R^i to be positive definite. In fact, in some design problems such as command tracking and model reduction, an objective function representing simply the tracking or model-matching errors does not include the control term, hence $R^i = 0$.

In this section, we provide analytical expressions for the evaluation of the objective function $J(t_f)$ and its gradients $\partial J/\partial C_o$ with respect to the controller matrix C_o. Details of the derivation can be found in [9]. For simple technicality, the problem formulation assumes that there is no implicit-loop paths within the feedback control system. Namely, the control input $u^i(t)$ or the measurement output $y_s^i(t)$ should not have any direct link to itself. This translates into the conditions that one of the product DD_{su}^i or $D_{su}^i D$ must be zero. This is not restrictive since in practice either actuation or sensor dynamics would be incorporated into the design models and thereby resulting in a system that satisfies the above assumption; or one can simply reformulate an equivalent problem with a set of measurement outputs where $D_{su}^i = 0$. Let's assume without loss of generality that we have the case of $DD_{su}^i = 0$, then the closed-loop system can be written in the form [9] shown in figure 2 or simply,

$$\dot{x}'^i(t) = F'^i x'^i(t) + \Gamma'^i \eta^i(t) \text{ with } x'^i(0) = 0 \quad (11)$$

where

$$[F'^i]_{(n+r+n')\times(n+r+n')} = \begin{bmatrix} F^i + G^i D H_s^i & G^i C & (\Gamma^i + G^i D D_{sw}^i) H_w^i \\ B(\mathrm{I} + D_{su}^i D) H_s^i & A + BC_{su}^i C & B(\mathrm{I} + D_{su}^i D) D_{sw}^i H_w^i \\ 0 & 0 & F_w^i \end{bmatrix} \quad (12)$$

$$[\Gamma'^i]_{(n+r+n')\times m'} = \begin{bmatrix} (\Gamma^i + G^i D D_{sw}^i) D_w^i \\ B(\mathrm{I} + D_{su}^i D) D_{sw}^i D_w^i \\ \Gamma_w^i \end{bmatrix} \quad (13)$$

$$[H_s'^i]_{p\times(n+r+n')} = \left[(\mathrm{I} + D_{su}^i D) H_s^i \; D_{su}^i C \; (\mathrm{I} + D_{su}^i D) D_{sw}^i H_w^i\right] \quad (14)$$

$$[D_s'^i]_{p\times m'} = \left[(\mathrm{I} + D_{su}^i D) D_{sw}^i D_w^i\right] \quad (15)$$

$$[H_c'^i]_{p'\times(n+r+n')} = \left[H_c^i + D_{cu}^i D H_s^i \; D_{cu}^i C \; (D_{cu}^i D D_{sw}^i + D_{cw}^i) H_w^i\right] \quad (16)$$

and

$$[C'^i]_{m\times(n+r+n')} = \left[D H_s^i \; C \; D D_{sw}^i H_w^i\right] \quad (17)$$

With these definitions, equations (2), (3) and (7) for $y_s^i(t), y_c^i(t)$ and $u^i(t)$ become

$$y_s^i(t) = H_s'^i x'^i(t) + D_s'^i \eta^i(t) \quad (18)$$

$$y_c^i(t) = H_c'^i x'^i(t) + (D_{cu}^i D D s w^i + D_{cw}^i) D_w^i \eta^i(t) \quad (19)$$

$$u^i(t) = C'^i x'^i(t) + D D_{sw}^i D_w^i \eta^i(t) \quad (20)$$

For a well-posed performance index $J(t_f)$, product of direct feedthrough term $(D_{cu}^i D D_{sw}^i + D_{cw}^i) D_w^i$ in the criterion output $y_c^i(t)$ and the penalty weighting matrix Q^i must be zero. Similarly, product of the direct feedthrough term $DD_{sw}^i D_w^i$ in the control output $u^i(t)$ and the penalty weighting matrix R^i must also be zero. Under these circumstances, the performance index $J(t_f)$ in equations (9) and (10) can be written as,

$$J(t_f) = \frac{1}{2} \sum_{i=1}^{N_p} W_p^i \, Trace\{L^i(t_f) \Gamma^i W_o^i \Gamma^{iT}\} \quad (21)$$

where

$$L^i(t_f) = \int_0^{t_f} e^{(F'+\alpha' I)t} [H_c^{iT} Q^i H_c^i + C^{iT} R^i C^i] e^{(F'+\alpha' I)^T t} dt \quad (22)$$

In the derivation of the analytical gradients of the performance index $J(t_f)$ with respect to design parameters in C_o of the controller state matrices, it is convenient to express the closed-loop system matrices in terms of C_o explicitly, as suggested in [9],

$$F'^i = F_o^i + (G_o^i + T_2 C_o D_1^i) C_o H_o^i \quad (23)$$

$$\Gamma'^i = \Gamma_o^i + (G_o^i + T_2 C_o D_1^i) C_o D_o^i \quad (24)$$

$$H_c'^i = H_1^i + D_2^i C_o H_o^i \quad (25)$$

$$C'^i = [DH_s^i \; C \; DD_{sw}^i H_w^i] = T_1 C_o H_o^i \quad (26)$$

where

$$[F_o^i]_{(n+r+n')\times(n+r+n')} = \begin{bmatrix} F^i & 0 & \Gamma^i H_w^i \\ 0 & 0 & 0 \\ 0 & 0 & F_w^i \end{bmatrix} \quad (27)$$

$$[G_o^i]_{(n+r+n')\times(m+r)} = \begin{bmatrix} G^i & 0 \\ 0 & \mathrm{I} \\ 0 & 0 \end{bmatrix} \quad (28)$$

$$[\Gamma_o^i]_{(n+r+n')\times m'} = \begin{bmatrix} \Gamma^i D_w^i \\ 0 \\ \Gamma_w^i \end{bmatrix} \quad (29)$$

$$[H_o^i]_{(p+r)\times(n+r+n')} = \begin{bmatrix} H_s^i & 0 & D_{sw}^i H_w^i \\ 0 & \mathrm{I} & 0 \end{bmatrix} \quad (30)$$

$$[H_1^i]_{p'\times(n+r+n')} = [H_c^i \; 0 \; D_{cw}^i H_w^i] \quad (31)$$

$$[D_o^i]_{(p+r)\times m'} = \begin{bmatrix} D_{sw}^i D_w^i \\ 0 \end{bmatrix} \quad (32)$$

$$[D_1^i]_{(p+r)\times(m+r)} = \begin{bmatrix} D_{su}^i & 0 \\ 0 & 0 \end{bmatrix} \quad (33)$$

$$[D_2^i]_{p'\times(m+r)} = [D_{cu}^i \; 0] \quad (34)$$

$$[T_1]_{m\times(m+r)} = [\mathrm{I} \; 0] \quad (35)$$

$$[T_2]_{(n+r+n')\times(m+r)} = \begin{bmatrix} 0 & 0 \\ 0 & \mathrm{I} \\ 0 & 0 \end{bmatrix} \quad (36)$$

$$[T_3]_{(p+r')\times(n+r+n')} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mathrm{I} & 0 \end{bmatrix} \quad (37)$$

It has been shown in [9] that derivative of the performance index $J(t_f)$ with respect to the controller matrix C_o (i.e $\partial\mathcal{J}/\partial C_o$) can be obtained explicitly from the following set of equations,

$$\partial\mathcal{J}/\partial C_o = \sum_{i=1}^{N_p} W_p^i \{(D_2^{i\mathrm{T}} Q H_c'^i + T_1^{\mathrm{T}} R C'^i)\mathcal{X}^i(t_f) H_o^{i\mathrm{T}} + (G_o^i + T_2 C_o D_1^i)^{\mathrm{T}}[\mathcal{M}^i(t_f) H_o^{i\mathrm{T}} + \mathcal{L}^i(t_f)\Gamma'^i W_o^i D_o^{i\mathrm{T}}] + T_2^{\mathrm{T}}[\mathcal{M}^i(t_f) H_o^{i\mathrm{T}} + \mathcal{L}^i(t_f)\Gamma'^i W_o^i D_o^{i\mathrm{T}}](D_1^i C_o)^{\mathrm{T}}]\} \quad (38)$$

where

$$\mathcal{X}^i(t_f) = \int_0^{t_f} e^{(F'^i+\alpha\mathrm{I})t}\,\Gamma'^i W_o^i \Gamma'^{i\mathrm{T}} e^{(F'^i+\alpha\mathrm{I})^{\mathrm{T}} t}\,dt \quad (39)$$

$$\mathcal{L}^i(t_f) = \int_0^{t_f} e^{(F'^i+\alpha\mathrm{I})^{\mathrm{T}} t}(H_c'^{i\mathrm{T}} Q H_c'^i + C'^{i\mathrm{T}} R C'^i) e^{(F'^i+\alpha\mathrm{I})t}\,dt \quad (40)$$

$$\mathcal{M}^i(t_f) = \int_0^{t_f}\int_0^t e^{(F'^i+\alpha\mathrm{I})^{\mathrm{T}}(t-\sigma)}(H_c'^{i\mathrm{T}} Q H_c'^i + C'^{i\mathrm{T}} R C'^i) e^{(F'^i+\alpha\mathrm{I})t}\Gamma'^i W_o^i \Gamma'^{i\mathrm{T}} e^{(F'^i+\alpha\mathrm{I})^{\mathrm{T}}\sigma}\,d\sigma\,dt \quad (41)$$

From equations (21)-(22) and (38)-(41), evaluation of the performance index $J(t_f)$ and its gradients $\partial\mathcal{J}/\partial C_o$ would require algorithms for efficient computation of the following two types of integrals of matrix exponentials: $\mathcal{X}(t) = \int_0^t e^{A\tau} B e^{C\tau}\,d\tau$ and $\mathcal{M}(t) = \int_0^t\int_0^v e^{A(v-s)} B e^{Cv} D e^{Es}\,ds\,dv$. In the next section, we review briefly the numerical algorithm developed in [9] for the computation of $\mathcal{X}(t)$ and $\mathcal{M}(t)$.

3 Current Method for Evaluating $\mathcal{X}(t)$ and $\mathcal{M}(t)$

Previous methods for evaluating $\mathcal{X}(t)$ and $\mathcal{M}(t)$ involve basically the diagonalization of the matrices A, C and E in the exponential functions. The procedure requires the determination of the eigenvalues and eigenvectors of these matrices. It is further assumed for convenience that similarity transformations can be constructed from these eigenvectors to diagonalize the respective matrices. Namely, there exist nonsingular transformations V_A and V_C and V_E such that

$$A = V_A \Lambda_A V_A^{-1}\,,\; C = V_C \Lambda_C V_C^{-1}\,,\; E = V_E \Lambda_E V_E^{-1} \quad (42)$$

where the matrices Λ_A, Λ_C and Λ_E are diagonal. Under these assumptions one can express for example the exponential function of e^{At} as

$$e^{At} = e^{V_A \Lambda_A V_A^{-1} t} = V_A e^{\Lambda_A t} V_A^{-1} \quad (43)$$

Usage of this decomposition in the calculation of $\mathcal{X}(t)$ is shown below.

$$\mathcal{X}(t) = \int_0^t e^{A\tau} B e^{C\tau}\,d\tau = V_A \left\{\int_0^t e^{\Lambda_A \tau}\mathcal{B} e^{\Lambda_C \tau}\,d\tau\right\} V_C^{-1} \quad (44)$$

where $\mathcal{B} = V_A^{-1} B V_C$. Advantage of this approach is based on the fact that the exponential function

of a diagonal matrix is also diagonal. In this case, time integration in $\mathcal{X}(t)$ can be performed directly by explicit integration of product of scalar exponential functions. The resulting numerical algorithm is quite accurate and efficient, provided that the transformation matrices V_A and V_C are not ill-conditionned. A similar procedure can also be applied to the evaluation of $\mathcal{M}(t)$. Complete discussion can be found in Appendices C and D of [9]. However, breakdown of this algorithm will occur when the matrices A , C or E become degenerate or near degenerate; a situation that becomes eminent when we address control of flexible structures with densely packed modes as demonstrated in the design examples of sections 8 and 9.

Clearly, in order to have a reliable design algorithm for optimal low-order output-feedback control synthesis [9], one must develop a robust numerical scheme to evaluate matrix integrals of the form shown in $\mathcal{X}(t)$ and $\mathcal{M}(t)$ for the case of a *degenerate* system.

4 Alternative Approaches for Solving $\mathcal{X}(t)$ and $\mathcal{M}(t)$

One rather simple approach is to evaluate

$$\mathcal{X}(t) = \int_0^t e^{A\tau} B e^{C\tau}\, d\tau \quad , \tag{45}$$

$$\mathcal{M}(t) = \int_0^t \int_0^v e^{A(v-s)} B e^{Cv} D e^{Es}\, ds\, dv \tag{46}$$

directly using techniques based on numerical quadrature. Efficiency of numerical integration techniques is poor; especially when it requires small integration step size for satisfactory accuracy in the case of stiff system matrices A, C and E. Another possibility is to use some types of algebraic Lyapunov equations for the solution of $\mathcal{X}(t)$ and $\mathcal{M}(t)$. For example, it can be easily shown that the matrix $\mathcal{X}(t)$ can be obtained from the solution of the following Lyapunov equation,

$$A\mathcal{X}(t) + \mathcal{X}(t)C = \left[e^{A\tau} B e^{C\tau}\right]_0^t \tag{47}$$

Solution of equation (47) exists if $\lambda_i(A) + \lambda_j(C) \neq 0$. This condition will not be satisfied in general for arbitrary system matrices A and C. Thus, from practical purposes $\mathcal{X}(t)$ and likewise $\mathcal{M}(t)$ cannot be solved from a scheme based on Lyapunov equations.

Another possible approach is based on the direct use of exponential matrix. It is well-known [12] that convolution integrals involving matrix exponentials, as represented in the matrices $\mathcal{X}(t)$ and $\mathcal{M}(t)$, can be derived from the matrix exponential of an augmented matrix. It can be shown that the matrix $\mathcal{X}(t)$ can be derived from a product of the following matrix exponentials,

$$\mathcal{X}(t) = e^{At}\begin{bmatrix} I & 0 \end{bmatrix} exp\left\{\begin{bmatrix} -A & B \\ 0 & C \end{bmatrix} t\right\}\begin{bmatrix} 0 \\ I \end{bmatrix} \tag{48}$$

Thus, computation of $\mathcal{X}(t)$ now involves the computation of a matrix exponential. A simple reliable algorithm for computing the matrix exponential is given in section 5.

In a similar fashion, one can express the matrix $\mathcal{M}(t)$ in terms of a submatrix of a matrix exponential. To see this, we start from its definition

$$\begin{aligned} \mathcal{M}(t) &= \int_0^t \int_0^v e^{A(v-s)} B e^{Cv} D e^{Es}\, ds\, dv \\ &= \int_0^t e^{-As}\left\{\int_s^t e^{Av} B e C v\, dv\right\} D e^{Es}\, ds \\ &= -\int_0^t e^{-As}\left\{\int_t^s e^{Av} B e^{Cv} dv\right\} \\ &\quad * D e^{Es}\, ds \end{aligned} \tag{49}$$

Let's perform a change of integration variable $v = t - r$. We have,

$$\begin{aligned} \mathcal{M}(t) &= \int_0^t e^{-As}\left\{\int_0^{t-s} e^{A(t-r)} B e^{C(t-r)}\, dr\right\} \\ &\quad * D e^{Es}\, ds \\ &= -\int_t^0 e^{A(t-s)}\left\{\int_0^{t-s} e^{-Ar} B e^{-Cr} dr\right\} \\ &\quad * e^{Ct} D e^{-E(t-s)} d(t-s) e^{Et} \\ &= \int_0^t e^{Aq}\left\{\int_0^q e^{-Ar} B e^{-Cr}\, dr\right\} \\ &\quad * e^{Ct} D e^{-Eq}\, dq\, e^{Et} \\ &= \int_0^t \left\{\int_0^q e^{A(q-r)} B e^{Ct/2} e^{-Cr}\, dr\right\} \\ &\quad * e^{Ct/2} D e^{E(t-q)} dq \end{aligned} \tag{50}$$

Notice that part of the integrand in equation (50) delimited by braces can be replaced by terms involving the exponential of an augmented matrix. This follows simply from results developed for the matrix

$\mathcal{X}(t)$. With this substitution, we obtain

$$\mathcal{M}(t) = \int_0^t \left\{ [I\ 0] \exp\left\{ \begin{bmatrix} A & Be^{Ct/2} \\ 0 & -C \end{bmatrix} q \right\} \begin{bmatrix} 0 \\ I \end{bmatrix} \right\} * e^{Ct/2} D e^{E(t-q)}\, dq \tag{51}$$

$$= \int_0^t [I\ 0] \exp\left\{ \begin{bmatrix} A & Be^{Ct/2} \\ 0 & -C \end{bmatrix} (t-q) \right\} * \left(\begin{bmatrix} 0 \\ I \end{bmatrix} e^{Ct/2} D \right) e^{Eq}\, dq$$

$$= [I\ 0\ 0] \exp\left\{ \begin{bmatrix} A & Be^{Ct/2} & 0 \\ 0 & -C & e^{Ct/2} D \\ 0 & 0 & E \end{bmatrix} t \right\} \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix}$$

In this section, we have shown that the matrices $\mathcal{X}(t)$ and $\mathcal{M}(t)$ can be formulated in terms of the solutions of some matrix exponentials. Their evaluation depends therefore strongly on the accuracy and reliability of numerical methods for computing matrix exponential. We will present one such algorithm in section 5. However for computational expediency, special consideration must also be taken to ensure the efficiency of the overall scheme when the upper limit t is large and one of the matrices A, C or D is unstable. Also one must economize memory requirements associated with high dimensionality of the augmented matrix when computing the matrix exponential. These considerations will be elaborated in sections 6 and 7 where we give precise algorithms for the computation of the matrices $\mathcal{X}(t)$ and $\mathcal{M}(t)$ respectively.

5 Numerical Method for the Matrix Exponential

Several numerical methods are available for the computation of the matrix exponential [11]. Among all these, an approximation method based on Padé series is found to be satisfactory [12]. An important component in any numerical routine for matrix exponential is the scaling of the matrix argument prior to the series calculation. Due to the simple result that $e^{At} = (e^{At/2})^2$, a scale factor in terms of powers of two (i.e 2^m) is often used. In this scheme, one can recover the actual value of the original matrix exponential by performing m squarings on the matrix exponential of the scaled matrix. The index m is determined based on the desired size for the scaled matrix. In our algorithm, scaling is applied to the original matrix until its ∞-norm $\|A\|_\infty$ falls below 1/2.

As mentioned above, the preferred series approximation in our computation of the matrix exponential is the Padé series. Let's review some of the unique features associated with the Padé series for the case of a scalar function $\mathcal{F}(z)$. On its most basic terms, it is a rational function of z of a preselected order that approximates the function $\mathcal{F}(z)$. For a given choice of the order of the numerator (say N) and of the denominator (say M), the Taylor series representation of this Padé series must match the power series representation of $\mathcal{F}(z)$ for the first (N+M+1) terms. Namely,

$$\mathcal{F}(z) \sim P_M^N(z) = \frac{\sum_{i=0}^N A_i z^i}{\sum_{i=0}^M B_i z^i} \tag{52}$$

In fact, the most common form of the Padé series is known as the diagonal sequence where the numerator and the denominator have the same order (i.e $M = N$). While it is known that the Padé series for the matrix exponential (i.e $\mathcal{F}(z) = e^z$) converges only slightly faster than the Taylor series for a scalar argument, the improvement is more significant for matrix argument. In the matrix case, Padé series involves computation of a numerator matrix $\mathcal{N}(At)$ and of a denominator matrix $\mathcal{D}(At)$. For a diagonal Padé series of order N, we have

$$\begin{aligned} \mathcal{D}(At) &= \mathrm{I} + \tfrac{(2N-1)!\,N!}{(2N)!\,(N-1)!} At \\ &+ \tfrac{(2N-2)!\,N!}{(2N)!\,2!\,(N-2)!} (At)^2 + \cdots \\ &+ \tfrac{(2N-i)!\,N!}{(2N)!\,i!\,(N-i)!} (At)^i + \cdots \\ &+ \tfrac{N!}{(2N)!} (At)^N \end{aligned} \tag{53}$$

and

$$\begin{aligned} \mathcal{N}(At) &= \mathrm{I} - \tfrac{(2N-1)!\,N!}{(2N)!\,(N-1)!} At \\ &+ \tfrac{(2N-2)!\,N!}{(2N)!\,2!\,(N-2)!} (At)^2 - \cdots \\ &+ (-1)^i \tfrac{(2N-i)!\,N!}{(2N)!\,i!\,(N-i)!} (At)^i + \cdots \\ &+ (-1)^N \tfrac{2N!}{(2N)!} (At)^N \end{aligned} \tag{54}$$

The matrix exponential is simply given by

$$e^{At} = \mathcal{D}^{-1}(At)\mathcal{N}(At) \tag{55}$$

Invertibility of $\mathcal{D}(At)$ is usually ensured by proper scaling of the matrix argument At.

Another important consideration in the Padé series is its length N. Assuming that the matrix At has been scaled such that $\|At\|_\infty$ is less than 1/2, the parameter N can be choosen according to [12] such that

$$2^{3-2N} \frac{(N!)^2}{(2N)!\,(2N+1)!} \leq \epsilon \tag{56}$$

where ϵ is a given desired tolerance for accuracy.

With a Padé series of N terms where N is determined from above, the approximation can be thought of as the exact calculation of a matrix exponential for a "nearby" matrix $(At+E)$ where E is the error matrix with $\|E\|_\infty \leq \epsilon\|At\|_\infty$. The relative error of the approximation is bounded by the following inequality,

$$\frac{\|e^{(At+E)} - e^{At}\|_\infty}{\|e^{At}\|_\infty} \leq \epsilon\|At\|_\infty e^{\epsilon\|At\|_\infty} \qquad (57)$$

Thus, reducing the ∞−norm of the matrix At would indeed improve the numerical accuracy of the matrix exponential. It has also been shown that methods by series approximation yield better accuracy if the matrix argument has been preconditioned. Additional improvement may therefore be gained by first preconditioning the original matrix. Another immediate benefit of lowering the ∞-norm of the matrix being exponentiated is that the actual scaling factor m needed would also be smaller; thereby resulting in a fewer number of matrix multiplications in the squaring procedure. As usual, preconditioning a matrix tends to bring singular values of that matrix closer together (i.e. lower the condition number), thus avoiding situation where scaling factor is predominantly determined by a few large singular values, and causing significant loss of precision related to the set of small singular values. The most common method used in the precondition of a matrix is the Osborne's method [14], which minimizes the Frobenius norm of that matrix (and thus indirectly lowering its ∞-norm). However, extensive tests conducted so far seems to indicate that preconditioning of a matrix did not yield significant reduction in the ∞-norm and a smaller scaling factor to justify the added computational efforts incurred in the Osborne's method. The procedure of preconditioning a matrix is nonetheless recommended from the point of view of improved accuracy (see [15] and[17]).

In the implementation of our design algorithm for optimal low-order controller synthesis [9], a value of $\epsilon = 10^{-8}$ has been selected requiring therefore a 4-terms Padé series (i.e $N = 4$) in the evaluation of the matrices $\mathcal{X}^i(t)$, $\mathcal{L}^i(t)$ and $\mathcal{M}^i(t)$ of equations (39)-(41). Additional considerations in the implementation of the proposed method for computing $\mathcal{X}(t)$ and $\mathcal{M}(t)$ are given in sections 6 and 7.

6 Detailed Algorithm for Computing $\mathcal{X}(t)$

As seen in the previous section, the matrix $\mathcal{X}(t)$ can be evaluated in terms of a matrix exponential as shown in equation (48). Conceptually, it is a simple and straightforward procedure to compute the matrix exponential of any arbitrary matrix using the Padé series discussed in section 5. However, it becomes a nontrivial task when we try to implement an efficient algorithm that examines carefully the issues related to accuracy, speed and memory requirement. The basic difficulties lie in the fact that the matrix exponential is for an augmented matrix of a particular form,

$$exp\left\{\begin{bmatrix} -A & B \\ 0 & C \end{bmatrix} t\right\} = \begin{bmatrix} e^{-At} & e^{-At}\mathcal{X}(t) \\ 0 & e^{Ct} \end{bmatrix} \qquad (58)$$

where $A = C^T = F^{ri} + \alpha^i I$ according to our problem in equations (39) and (40) for the matrices $\mathcal{X}^i(t_f)$ and $\mathcal{L}^i(t_f)$ respectively. Clearly, if the system matrix A is stable (i.e all the eigenvalues of the matrix A have negative real parts) then one could easily encounter numerical overflow when evaluating the term e^{-At} even though the matrix integrals $\mathcal{X}(t)$ and $\mathcal{L}(t)$ of interest are perfectly well-behaved. The overflow problem occurs most likely in the final squaring process. To arrive at a feasible approach in the evaluation of $\mathcal{X}(t)$, one needs to examine in details the steps taken in arriving at the matrix exponential of the original matrix starting from that of a scaled matrix (i.e in the squaring process).

Let's assume that one has scaled the input matrix A by $A\Delta t$ where Δt is a reasonably small time interval given by $\Delta t = t/n = t/2^m$. Thus, we need to first evaluate

$$exp\left\{\begin{bmatrix} -A & B \\ 0 & C \end{bmatrix} \Delta t\right\} \text{ where } \Delta t = t/n = t/2^m.$$

For notation convenience, we define

$$exp\left\{\begin{bmatrix} -A & B \\ 0 & C \end{bmatrix} \Delta t\right\} = \begin{bmatrix} D & E \\ 0 & F \end{bmatrix}$$

$$= \begin{bmatrix} e^{-A\Delta t} & e^{-A\Delta t}\int_0^{\Delta t} e^{A\tau} B e^{C\tau}\, d\tau \\ 0 & e^{C\Delta t} \end{bmatrix} \qquad (59)$$

Furthermore, let $W = exp(A\Delta t) = D^{-1}$. Now we can write our result as follows,

$$\begin{aligned} \mathcal{X}(t) &= W^n[D^{n-1}E + D^{n-2}EF \\ &\quad + D^{n-3}EF^2 + \cdots + EF^{n-1}], \end{aligned} \qquad (60)$$

or

$$\mathcal{X}(t) = W[E + WEF + W^2EF^2 + \cdots + W^{n-1}EF^{n-1}]. \qquad (61)$$

The above results are produced by performing m squarings of $\begin{bmatrix} D & E \\ 0 & F \end{bmatrix}$ and taking the appropriate submatrix for $\mathcal{X}(t)$. In our application (cf. equations

(39)-(40)), the solution would therefore involve products of matrices of size $2(n+r+n')$. Close examination of equation (61) leads to the following algorithm involving only product of matrices of size $(n+r+n')$ with the final result achievable in m steps,

Step 1 :
$P_1 = W, \quad Q_1 = E, \quad R_1 = F$

Step 2 :
$P_2 = P_1^2, \quad Q_2 = Q_1 + P_1 Q_1 R_1, \quad R_2 = R_1^2$
$\cdots = \cdots, \quad \cdots = \cdots, \quad \cdots = \cdots$

Step m :
$P_m = P_{m-1}^2 \quad Q_m = Q_{m-1} + P_{m-1} Q_{m-1} R_{m-1}, \quad R_m = R_{m-1}^2$

Finally, $\mathcal{X}(t) = WQ_m$. It should be noted that one can "absorb" this extra factor of W ($= e^{A\Delta t}$) into the matrix Q_1 without any change to the above algorithm (i.e starting the above algorithm with $Q_1 = WE$ instead). This removes the need to retain the matrix W throughout the computation.

Finally, one notes that the terms P_i or R_i for $(i = 1, m)$ may underflow and become a null matrix for some i; in particular when the scaling factor is large (i.e m large). When this situation happens, one can simply truncate the series calculation for $\mathcal{X}(t)$ up to the i^{th} step in the above algorithm since all of the significant (and nonzero) terms have already been accumulated into the matrix Q_i.

7 Detailed Algorithm for Computing $\mathcal{M}(t)$

Here the numerical algorithm is a bit involved compared to the one given for the calculation of $\mathcal{X}(t)$. This is largely due to the increased complexity of the argument of the matrix exponential. Following the procedure described in section 6, let's perform a scaling upon the input matrix A by $A\Delta t$ such that computation of the matrix quantities $\mathcal{M}_o, H, J, P, U$ and $W = V^{-1}$ is well-behaved. These quantities are defined from the following matrix exponential,

$$exp\left\{\begin{bmatrix} A & Be^{Ct/2} & 0 \\ 0 & -C & e^{Ct/2}D \\ 0 & 0 & E \end{bmatrix}\Delta t\right\} = \begin{bmatrix} P & He^{Ct/2} & \mathcal{M}_o \\ 0 & V & e^{Ct/2}J \\ 0 & 0 & U \end{bmatrix} \tag{62}$$

Due to the possible numerical underflow in the matrix $e^{Ct/2}$ for large t, the matrices H and J are computed directly from the following definitions,

$$H = \int_0^{\Delta t} e^{A\tau} B e^{C\tau}\, d\tau e^{-C\Delta t} \tag{63}$$

and

$$J = e^{-C\Delta t} \int_0^{\Delta t} e^{C\tau} D e^{E\tau}\, d\tau \tag{64}$$

However, the computation of $\mathcal{M}_o$ in equation (62) can still underflow due to its explicit dependence on $e^{Ct/2}$. For the calculation of the matrix $\mathcal{M}(t)$, ideally it can be obtained from m squarings of equation (62). If carried out in this manner, potential numerical overflow is eminent since, according to our equation for $\mathcal{M}^i(t_f)$ in (41), we have $A^T = C = E^T = F^i + \alpha^i I$. Hence, if the matrix C is stable, then the matrix exponential $e^{-Ct} = V^n$ will become unbounded. To bypass this difficulty, as in the calculation for $\mathcal{X}(t)$, one needs to conduct the squaring algorithm explicitly. It can be shown that the matrix $\mathcal{M}(t)$ can be computed as

$$\begin{aligned} \mathcal{M}(t) &= P^{n-1}\mathcal{M}_o + P^{n-2}\mathcal{M}_o U \\ &+ P^{n-3}\mathcal{M}_o U^2 + \cdots + P\mathcal{M}_o U^{n-2} \\ &+ \mathcal{M}_o U^{n-1} \\ &+ HW^2J + HW^3JU + PHW^3J \\ &+ PHW^4JU + P^2HW^4J + \cdots \\ &+ P^{n-2}HW^nJ \end{aligned} \tag{65}$$

This formulation no longer involves the matrix V. The above series for $\mathcal{M}$ can be distinguished into two parts—one that contains the matrix $\mathcal{M}_o$ and the other that does not. The terms involving $\mathcal{M}_o$ can be thought of as

$$[I\ 0]\begin{bmatrix} P & \mathcal{M}_o \\ 0 & U \end{bmatrix}^n \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{66}$$

which can be performed by m squarings. The remaining terms involving H, J, W, P, and U are computationally intensive and are of the form

$$\sum_{i=0}^{n-2}\sum_{j=0}^{n-2} P^i H W^{2+i+j} J U^j, \quad \text{where } 2+i+j \le n. \tag{67}$$

This equation, owing to the restriction $2+i+j \le n$, is not easily calculated in $\mathbf{m}(= log_2(n))$ steps. A reasonably efficient procedure for computing the final matrix $\mathcal{M}(t)$ is to merge both the easily computed portion given in equation (66) and the more difficult series in equation (67) into a sequence of m steps, as shown in figure 3. Due to potential numerical underflow, the term W^{i-2} is not accurately obtained from the product $W^i V^2$ where $V = W^{-1}$. Indeed one needs to recompute the term W^{n+2-2^j} at each step

of the above algorithm. This could become the major drawback of our scheme even though we have used an efficient matrix exponentiation routine to compute W^i requiring at most $2 * log_2(i)$ matrix multiplies.

If in addition W^n is zero (or effectively so), restriction on the indices i and j of $2+i+j \leq n$ in equation (67) becomes inconsequential; hence we can express

$$\begin{aligned}\sum_{i=0}^{n-2}\sum_{j=0}^{n-2} P^i H W^{2+i+j} J U^j \;=\; & \\ (H + PHW + P^2HW^2 + \cdots)W^2 & \\ *(J + WJU + W^2JU^2 + \cdots) &\end{aligned}$$

resulting in a simpler algorithm involving the following three terms:

$$(a)\ [I\ 0]\begin{bmatrix} P & \mathcal{M}_o \\ 0 & U \end{bmatrix}^n \begin{bmatrix} 0 \\ I \end{bmatrix}$$

$$(b)\ (H + PHW + P^2HW^2 + \cdots + P^{n-2}HW^{n-2})$$

$$(c)\ (J + WJU + W^2JU^2 + \cdots + W^{n-2}JU^{n-2})$$

This algorithm can again be computed in m steps as seen in figure 4, but now there is no costly evaluation of a W^k term at each step.

Further simplification of the above algorithm can be achieved if we make use of the fact that we have $A = E = C^T$ (cf. equation (41)) and therefore $U = P = W^T$. If, for some index $j < m$, W^{2^j} (and likewise U^{2^j} and P^{2^j}) is zero or nearly so, then this calculation for $\mathcal{M}(t)$ is reduced to $\mathcal{M}(t) = H_j W^2 J_j$ since $M_m = 0$, $H_j = H_m$, and $J_j = J_m$.

In the following sections, we compare the usefulness of the proposed algorithm to the early algorithm presented in [9] in the design of low-order optimal controllers for two flexible mechanical systems .

8 A Simple Two-Mass-Spring Design Problem

Control of flexible mechanical systems has been of interest in recent years [18]. This problem provides us a simple design case where degeneracy in the closed-loop eigensystem can be easily illustrated. The problem is to control the displacement of the second mass by applying a force to the first mass as shown in Figure 5. At the start, it is simple to verify that the basic open-loop system has a pair of degenerate eigenvalues at the origin. Equations for the dynamic model are given below,

$$\begin{aligned} m_1\ddot{y_1} &= k(y_2 - y_1) + u + w \\ m_2\ddot{y_2} &= k(y_1 - y_2) \end{aligned} \tag{68}$$

or

$$\frac{d}{dt}\begin{bmatrix} y_1 \\ y_1' \\ y_2 \\ y_2' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -k/m_1 & 0 & k/m_1 \\ 0 & 0 & 1 & 0 \\ 0 & k/m_2 & 0 & -k/m_2 \end{bmatrix}\begin{bmatrix} y_1 \\ y_1' \\ y_2 \\ y_2' \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m_1 \\ 0 \\ 0 \end{bmatrix}(u+w) \tag{69}$$

where $m_1 = m_2 = k_1 = k_2 = 1$. For comparison, we have obtained an optimal second-order controller design of the form,

$$A = \begin{bmatrix} 0 & 1 \\ A_{21} & A_{22} \end{bmatrix};\quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$C = [C_{11}\ C_{12}];\qquad D = [D_{11}]$$

using both algorithms. The control design problem is to minimize the following H^2-norm of the closed-loop transfer function $T_{y_2 w}$ between the disturbance w and the displacement y_2 of the second mass through the controller design parameters $A_{21}, A_{22}, C_{11}, C_{12}$ and D_{11}. We start with the following *arbitrary* initial design guess of $A_{21} = -2, A_{22} = -1, C_{11} = 0, C_{12} = 0.5$, and $D_{11} = 0$. Both algorithms converge effectively to the same optimal design gains of $A_{21} = -0.8571, A_{22} = -0.9258, C_{11} = 0, C_{12} = -0.4535$ and $D_{11} = -0.2449$ and with an optimum value $\|T_{y_2 w}\|_2^2 = 7.71838215122$. A summary of the resulting closed-loop eigenvalues is given in Table 1. The main difference between the two algorithms is in the CPU time for the overall computation. Results are obtained for a VAX/VMS-Workstation DEC-3500 as follows: CPU time of $19.59 sec$ with the algorithm based on diagonalization and $97.36 sec$ using the proposed method. The increase in computational load is expected and constitutes the basic trade-off between reliability and speed of the solution algorithm. The proposed algorithm is more reliable and with this advantage does take a bit longer in computational time. With the early algorithm of [9], one cannot initiate the search for an optimal compensator design with zero gains (i.e $A_{21} = A_{22} = C_{11} = C_{12} = D_{11} = 0$) because, in this case, the closed-loop system would have two pairs of degenerate eigenvalues at the origin; one for the rigid-body mode and the other from the open-loop compensator poles. To alleviate this problem, it was suggested that one simply starts with any compensator design (stabilizing or not) that produces initially a non-degenerate closed-loop system. Even with these considerations, it was found that occasionally the algorithm could break down due to the

presence of near degeneracies in the closed-loop system matrix. Thus, for a reliable design method, solution algorithm must treat degeneracies as a common occurrence. This situation is more evident in the optimal output-feedback control design for high-order structural models with closely packed flexible modes.

Future consideration would be to develop a hybrid algorithm taking advantage of the computational efficiency of diagonalization when the closed-loop system matrix is not degenerate, and turning to the current algorithm when degeneracies are detected. System degeneracy can be easily checked from the condition number of the eigenvector matrix.

In the next section, we present a design problem where degeneracies occur frequently and therefore it could pose a serious difficulty for the early design algorithm based on diagonalization.

9 The JPL Large Space Structure Control Design

In control problems for large flexible mechanical systems such as space structures, causes of eigenvalue degeneracies are usually more subtle in nature than the simple case presented in section 8 for a two-mass-spring system. The JPL large space structure has been carefully designed to simulate a lightweight, non-rigid and lightly damped structure in a weightless environment [16]. The structure itself resembles a large antenna with a central boom-dish apparatus and an extended dish consisted of hoop wires and 12 ribs (Figure 6). There are two torque actuators (labelled $HA1$ and $HA10$) on the boom and dish structure to control the two angular degrees of freedom in pointing maneuver, and force actuators at four rib root locations (labelled $RA1$, $RA4$, $RA7$ and $RA10$) for vibration control. From the point of view of control design, it is a challenging problem since the plant has many closely spaced modes and is of reasonably high order. There are a total of 30 modes in the basic structural model. The flexible modes are lightly damped with damping ratios ranging from 0.007 to 0.01. The two rigid-body modes have a damping ratio of 0.12. Our design concept is to use two available angular displacement sensors $HS1$ and $HS10$ of the boom-dish apparatus and the two torquers $HA1$ and $HA10$ collocated with these sensors for control synthesis. With this selection, 20 of the flexible modes associated primarily with the rib motion become uncontrollable and unobservable. These modes are removed by modal truncation from our plant synthesis model. Eigenvalues of the remaining 10 modes are shown in Table 2.

An optimal low-order controller is designed to dampen vibration of the antenna to external excitations. To evaluate the effectivenes of the control system, we perform the following test. The entire structure is agitated using the two boom-dish actuators for the first 6.4 seconds with an applied torque in the form of a square wave of 0.8 second in width and with an amplitude of 1 N-m. The control system is then activated right after the excitation has been removed, and responses of the excited structure at the sensors are examined. The design objective is to damp out the induced vibration as fast as possible without excessive use of controls. Note that the natural responses of the structure will take about a few minutes to decay to zero (Figure 8).

For practical implementation, the controller design is choosen to be of 6^{th} order and has the following form,

$$A = \begin{bmatrix} -50 & 0 & A_{13} & A_{14} & A_{15} & A_{16} \\ 0 & -50 & A_{23} & A_{24} & A_{25} & A_{26} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & A_{43} & A_{44} & 0 & 0 \\ 0 & 0 & A_{53} & A_{54} & 0 & 1 \\ 0 & 0 & A_{63} & A_{64} & A_{65} & A_{66} \end{bmatrix} \tag{70}$$

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ B_{31} & B_{32} \\ B_{41} & B_{42} \\ B_{51} & B_{52} \\ B_{61} & B_{62} \end{bmatrix}$$

$$C = \begin{bmatrix} 50 & 0 & 0 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The first two states in the controller model serve as roll-filters, limiting the control bandwidth to less than $50 rad/sec$. In the design optimization, we have a total of 28 design variables: 16 in the controller A matrix and 12 in the B matrix. The objective function for design optimization consists of a sum of weighted H^2-norms of physical response variables observed at different location of the structure. It is of the form

$$J(t_f) = \lim_{t_f \to \infty} \frac{1}{2} \left\{ \sum_{i=1}^{12} Q_i E_\alpha \left[y_i^2(t_f) \right] + \sum_{j=1}^{2} R_j E_\alpha \left[u_j^2(t_f) \right] \right\} \tag{71}$$

Note that the expectation operator $E_\alpha[-]$ is for a system destabilized by a factor α. Table 3 lists the

design variables y_i and their corresponding penalty weightings Q_i. Also given in the table are the control design weightings R_j for the actuators $HA1$ and $HA10$. Responses in the above objective function are evaluated to random disturbances of unit white-noise spectra applied simultaneously at *all* the hub and rib actuators.

The design optimization begins with the following ***arbitrary*** initial guess on the controller matrices A and B,

$$A = \begin{bmatrix} -50 & 0 & 1 & 0 & 0 & 0 \\ 0 & -50 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -4 & -4 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$$

A destabilization factor α of 0.071 was used to ensure that all the closed-loop eigenvalues have a real part less than -0.071. The optimization fails to converge when a destabilization factor of greater than 0.075 was selected. This difficulty seems to be in moving the modes at 1.68 Hz under this controller configuration, implying that additional degrees of freedom must be added to the controller structure given in equation (70).

While the optimization convergence itself took 13.5 hours on a VAX/VMS Workstation DEC-3500, the proposed algorithm for the calculation of the objective function and its gradients with respect to the design parameters is robust and leads to well-behaved design convergence. The final optimal values of the A and B matrices are shown in Figure 7. Closed-loop eigenvalues are given in Table 4. Primary improvement is seen in the increased damping of two modes at 0.65 Hz.

Closed-loop responses of the sensor and control variables corresponding to this design are shown in Figure 8. The controlled responses decay to zero in about $20sec$ after the excitation has been removed. Notice that the control torques are within the desired limits of 1 N-m; the results are obtained through adjustment of the control design weights R_j in Table 3. This design example demonstrates the usefulness of a design algorithm for robust low-order controllers using parameter optimization, and the accompanying improvement of solution reliability using the algorithms described in sections 6 and 7 for degenerate systems.

10 Conclusions

Numerical algorithms for computing matrix exponentials and integrals of matrix exponentials have been developed to handle cases where the system matrix is degenerate. Numerical optimization combined with the given algorithms for the evaluation of the cost function and its gradients with respect to the controller design parameters has well-behaved convergence even when the closed-loop system becomes degenerate. These algorithms have been incorporated into a computer-aided-design package for synthesizing optimal output-feedback controllers. Reliability of the algorithm has been demonstrated using typical design problems encountered in the control of flexible structures. Clearly this algorithm when combined with a previous one based on diagonalization would enhance significantly the overall reliability of the optimal design procedure for low-order controllers, thereby providing an effective automated design environment for multivariable control synthesis.

References

[1] Stein, G. and Athans, M., "The LQG / LTR Procedure for Multivariable Feedback Control Design," *IEEE Transactions on Automatic Control*, AC-32, pp.105-114, 1987.

[2] Doyle,J. C., Glover, K. , Khargonekar, P. P. and Francis, B. A., "State-Space Solutions to Standard H_2 and H_∞-Control Problems," *IEEE Transactions on Automatic Control*, Vol. AC-34, No. 8, pp. 831-847, 1989.

[3] Doyle, J. C. and Glover, K., "State-Space Formulae for All Stabilizing Controllers that Satisfy an H_∞-Norm Bound and Relations to Risk Sensitivity," *Systems & Control Letters*, 11, pp. 167-172, 1988.

[4] Stoorvogel, A. A., "The Singular Control Problem with Dynamic Measurement Feedback," *SIAM J. Control and Optimization*, Vol. 29, No. 1, pp. 160-184, 1991.

[5] Stoorvogel, A. A. and Trentelman, H. L., "The Quadratic Matrix Inequality in Singular H_∞-Control with State Feedback," *SIAM J. Control and Optimization*, Vol. 28, No. 5, pp.1190-1208, September 1990.

[6] Levine, W.S. and Athans, M.,"On the Determination of the Optimal Constant Output Feedback Gains for Linear Multivariable Systems," *IEEE Transactions on Automatic Control*, AC-15, pp.44-48, 1970.

[7] Anderson, B. D. O. and Moore, J. B., *Linear Optimal Control*, Englewood Cliffs, Prentice Hall, Inc., New Jersey, 1971.

[8] Makila, P. M., "Computational Methods for Parametric LQ Problems–A Survey," *IEEE Transactions on Automatic Control*, AC-32, No.8, pp.658-671, August 1987.

[9] Ly, U. *A Design Algorithm for Robust Low-Order Controllers*, PhD. Thesis, Department of Aeronautics and Astronautics, Stanford University, November, 1982.

[10] Gill, P. E., Murray, W., Saunders, M. A. and Wright, M. H.,*User's Guide for NPSOL (Version 4.0): A Fortran Package for Nonlinear Programming.* Technical Report SOL-86-2, Stanford university, January 1986.

[11] VanLoan, C. F., "Nineteen Dubious Ways to Compute the Exponential of a Matrix", *SIAM Review* 20, pp.801-836, 1978.

[12] Golub, G. H. and VanLoan, C. F., *Matrix Computations.* Johns Hopkins University Press, pp 396-400, 1985.

[13] Bender, C. M. and Orzag, S. A. *Advanced Mathematical Methods for Scientists and Engineers.* McGraw-Hill 1978. pp 383-389.

[14] Osborne, E. E. , "On Preconditioning of Matrices", *Journal of the Association for Computing Machinery*, Los Angeles, CA. pp 338-345. March, 1960.

[15] Parlett, B. N. and Reinsch, C., "Balancing a Matrix for Calculation of Eigenvalues and Eigenvectors", *Numerical Math*, 13, pp. 293-304 , 1969 .

[16] Vivian, H.C. , Blaire, P.E., Eldred, D.B., Fleischer, G.E., Ih, C.-H.C., Nerheim, N.M., Scheid, R.E. and Wen, J.T., *Flexible Structure Control Laboratory Development and Technology Demonstration*, JPL Publication 88-29, October 1, 1987.

[17] Westreich, D. , "A Practical Method for Computing the Exponential of a Matrix and its Integral", *Communications in Applied Numerical Methods*, Vol 6, pp. 375-380, 1990.

[18] Wie, B. and Bernstein, D.,"A Benchmark Problem for Robust Control Design," *Proceedings of the 1990 American Control Conference*, May 23-25, 1990.

Eigenvalue			Damping	Freq (Hz)
-0.2290	±	0.3397i	0.559	0.0652
-0.1553	±	0.8480i	0.180	0.1372
-0.0786	±	1.2950i	0.061	0.2065

Table 1: Closed-Loop Modes (2 Mass-Spring System)

Eigenvalue			Damping	Freq (Hz)
-0.09500	±	0.7860i	0.120	0.12600
-0.08575	±	0.7093i	0.120	0.13704
-0.02802	±	4.0024i	0.007	0.63701
-0.02929	±	4.1844i	0.007	0.66598
-0.07405	±	10.583i	0.007	1.68434
-0.07405	±	10.583i	0.007	1.68434
-0.11310	±	10.616i	0.007	2.57123
-0.11785	±	16.384i	0.007	2.67929
-0.21365	±	30.520i	0.007	4.85749
-0.21365	±	30.520i	0.007	4.85749

Table 2: Open-Loop Modes of Antenna Structure

Variable	Q_i	Description
$RS1$	4100	Rib #1 root velocity
$RS4$	3950	Rib #4 root velocity
$RS7$	3975	Rib #7 root velocity
$RS10$	4050	Rib #10 root velocity
$HS1$	16500	Hub angular velocity
$HS10$	15600	Hub angular velocity
RS1	1100	Rib #1 root displacement
RS4	1050	Rib #4 root displacement
RS7	1150	Rib #7 root displacement
RS10	1025	Rib #10 root displacement
HS1	3900	Hub angular displacement
HS10	4100	Hub angular displacement
Variable	R_i	Description
HA1	41	Hub torque actuator
HA10	40	Hub torque actuator

Table 3: Design Variables for Antenna Structure

Eigenvalue			Damping	Freq (Hz)
-0.086899	±	0.6588i	0.1308	0.1058
-0.089071	±	0.7410i	0.1193	0.1188
-0.3165	±	3.624i	0.0870	0.5790
-0.2528	±	3.790i	0.0666	0.6045
-0.2162	±	4.112i	0.0525	0.6553
-0.2056	±	4.185i	0.0491	0.6669
-0.074193	±	10.58i	0.0070	1.684
-0.074589	±	10.58i	0.0070	1.684
-0.1168	±	16.15i	0.0072	2.570
-0.1253	±	16.83i	0.0074	2.678
-0.2142	±	30.52i	0.0070	4.857
-0.2143	±	30.52i	0.0070	4.857
-49.99			1.000	7.956
-49.99			1.000	7.956

Table 4: Boom-Dish-Controller Closed-Loop Modes

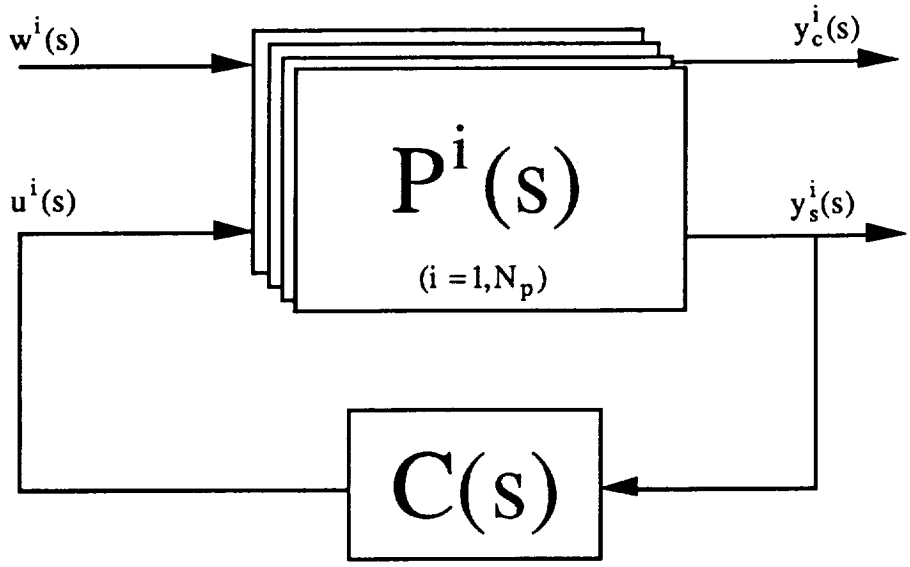

Figure 1: A Typical Closed-Loop System with a Feedback/Feedforward Controller

$$\begin{bmatrix} \dot{x}^i(t) \\ \dot{z}(t) \\ \dot{x}^i_w(t) \end{bmatrix} = \begin{bmatrix} F^i + G^i D H^i_s & G^i C & (\Gamma^i + G^i D D^i_{sw}) H^i_w \\ B(\mathrm{I} + D^i_{su} D) H^i_s & A + B C^i_{su} C & B(\mathrm{I} + D^i_{su} D) D^i_{sw} H^i_w \\ 0 & 0 & F^i_w \end{bmatrix} \begin{bmatrix} x^i(t) \\ z(t) \\ x^i_w(t) \end{bmatrix} + \begin{bmatrix} (\Gamma^i + G^i D D^i_{sw}) D^i_w \\ B(\mathrm{I} + D^i_{su} D) D^i_{sw} D^i_w \\ \Gamma^i_w \end{bmatrix} \eta^i(t)$$

Figure 2: State Model of the Closed-Loop System

Step 0 :					
$\mathcal{M}_o$	H	J	P	U	W
Step 1 :					
$\mathcal{M}_1 = P\mathcal{M}_o + \mathcal{M}_o U$	$H_1 = H$	$J_1 = J$	P^2	U^2	W^2
$+HW^n J$	$+PHW$	$+WJU$			
Step 2 :					
$\mathcal{M}_2 = P^2\mathcal{M}_1 + \mathcal{M}_1 U^2$	$H_2 = H_1$	$J_2 = J_1$	P^4	U^4	W^4
$+H_1 W^{n-2} J_1$	$+P^2 H_1 W^2$	$+W^2 J_1 U^2$			
Step 3 :					
$\mathcal{M}_3 = P^4\mathcal{M}_2 + \mathcal{M}_2 U^4$	$H_3 = H_2$	$J_3 = J_2$	P^8	U^8	W^8
$+H_2 W^{n-6} J_2$	$+P^4 H_2 W^4$	$+W^4 J_2 U^4$			
...	...	...	...	...	...
Step j :					
$\mathcal{M}_j = P^{2^{j-1}} M_{j-1} + M_{j-1} U^{2^{j-1}}$	$H_j = H_{j-1}$	$J_j = J_{j-1}$	P^{2^j}	U^{2^j}	W^{2^j}
$+H_{j-1} W^{n+2-2^j} J_{j-1}$	$+P^{2^{j-1}} H_{j-1} W^{2^{j-1}}$	$+W^{2^{j-1}} J_{j-1} U^{2^{j-1}}$			
...	...	...	...	...	...
Step m :					
$\mathcal{M}_m = P^{n/2} M_{m-1} + M_{m-1} U^{n/2}$	$H_m = H_{m-1}$	$J_m = J_{m-1}$	P^n	U^n	W^n
$+H_{m-1} W^2 J_{m-1}$	$+P^{n/2} H_{m-1} W^{n/2}$	$+W^{n/2} J_{m-1} U^{n/2}$			
$\mathcal{M}(t) = \mathcal{M}_m$					

Figure 3: An m-Step Calculation of $\mathcal{M}(t)$

Step 0 : $\bar{\mathcal{M}}_o$	H	J	U	P	W
Step 1 : $\bar{\mathcal{M}}_1 = P\bar{\mathcal{M}}_o + \bar{\mathcal{M}}_o U$	$H_1 = H$ $+PHW$	$J_1 = J$ $+WJU$	U^2	P^2	W^2
Step 2 : $\bar{\mathcal{M}}_2 = P^2\bar{\mathcal{M}}_1 + \bar{\mathcal{M}}_1 U^2$	$H_2 = H_1$ $+P^2 H_1 W^2$	$J_2 = J_1$ $+W^2 J_1 U^2$	U^4	P^4	W^4
Step 3 : $\bar{\mathcal{M}}_3 = P^4\bar{\mathcal{M}}_2 + \bar{\mathcal{M}}_2 U^4$	$H_3 = H_2$ $+P^4 H_2 W^4$	$J_3 = J_2$ $+W^4 J_2 U^4$	U^8	P^8	W^8
...	...	...	...	...	...
Step j : $\bar{\mathcal{M}}_j = P^{2^{j-1}}\bar{\mathcal{M}}_{j-1}$ $+\mathcal{M}_{j-1}U^{2^{j-1}}$	$H_j = H_{j-1}$ $+P^{2^{j-1}} H_{j-1} W^{2^{j-1}}$	$J_j = J_{j-1}$ $+W^{2^{j-1}} J_{j-1} U^{2^{j-1}}$	U^{2^j}	P^{2^j}	W^{2^j}
...	...	...	...	...	...
Step m : $\bar{\mathcal{M}}_m = P^{n/2}\bar{\mathcal{M}}_{m-1}$ $+\bar{\mathcal{M}}_{m-1}U^{n/2}$	$H_m = H_{m-1}$ $+P^{n/2} H_{m-1} W^{n/2}$	$J_m = J_{m-1}$ $+W^{n/2} J_{m-1} U^{n/2}$	U^n	P^n	W^n
$\mathcal{M} = \bar{\mathcal{M}}_m + H_m W^2 J_m$					

Figure 4: A Simplified m-Step Calculation of $\mathcal{M}(t)$

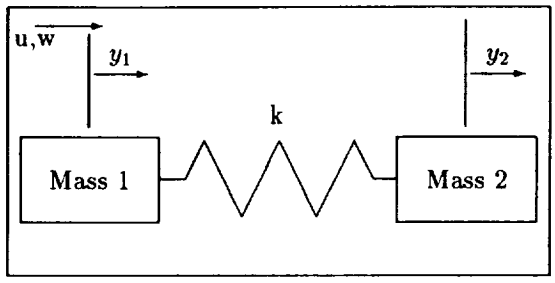

Figure 5: A Two-Mass-Spring Mass System

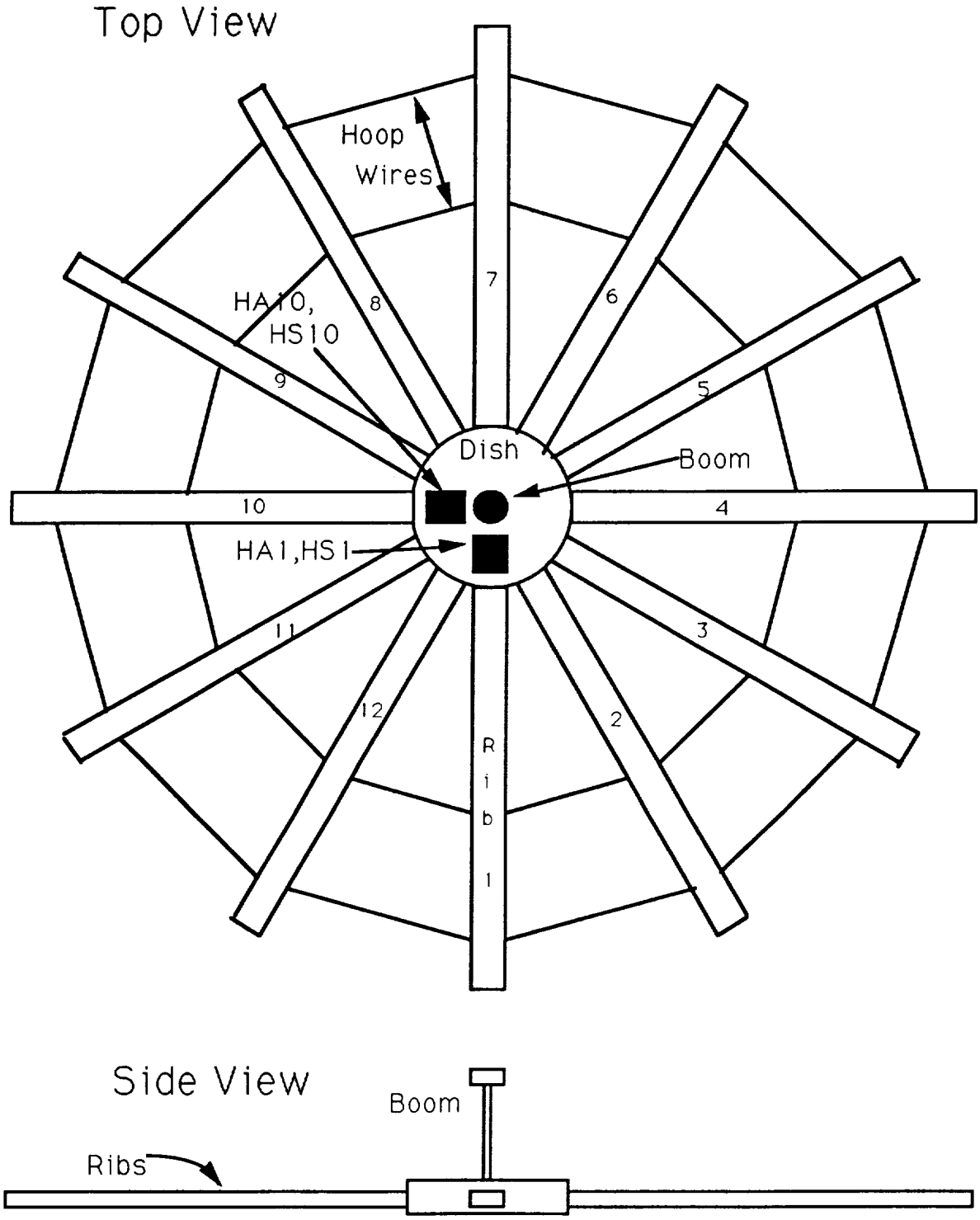

Figure 6: Antenna Structure

$$A = \begin{bmatrix} -50 & 0 & 2.874 & -2.270 & 1.7322 \times 10^{-3} & -2.2131 \times 10^{-4} \\ 0 & -50 & 1.225 & 0.7825 & 6.551 & -1.037 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -15.73 & -0.8799 & 0 & 0 \\ 0 & 0 & 1.560 & 0.2256 & 0 & 1 \\ 0 & 0 & 2.400 & -1.269 & -13.62 & -0.9810 \end{bmatrix}$$

$$B = \begin{bmatrix} 5.343 & -1.2310 \times 10^{-4} \\ 6.2118 \times 10^{-4} & 4.783 \\ 2.701 & -8.1595 \times 10^{-4} \\ 2.221 & 9.3152 \times 10^{-4} \\ -0.5147 & 5.379 \\ 1.614 & -1.252 \end{bmatrix}$$

Figure 7: Optimized Controller Matrices for LSCL Problem

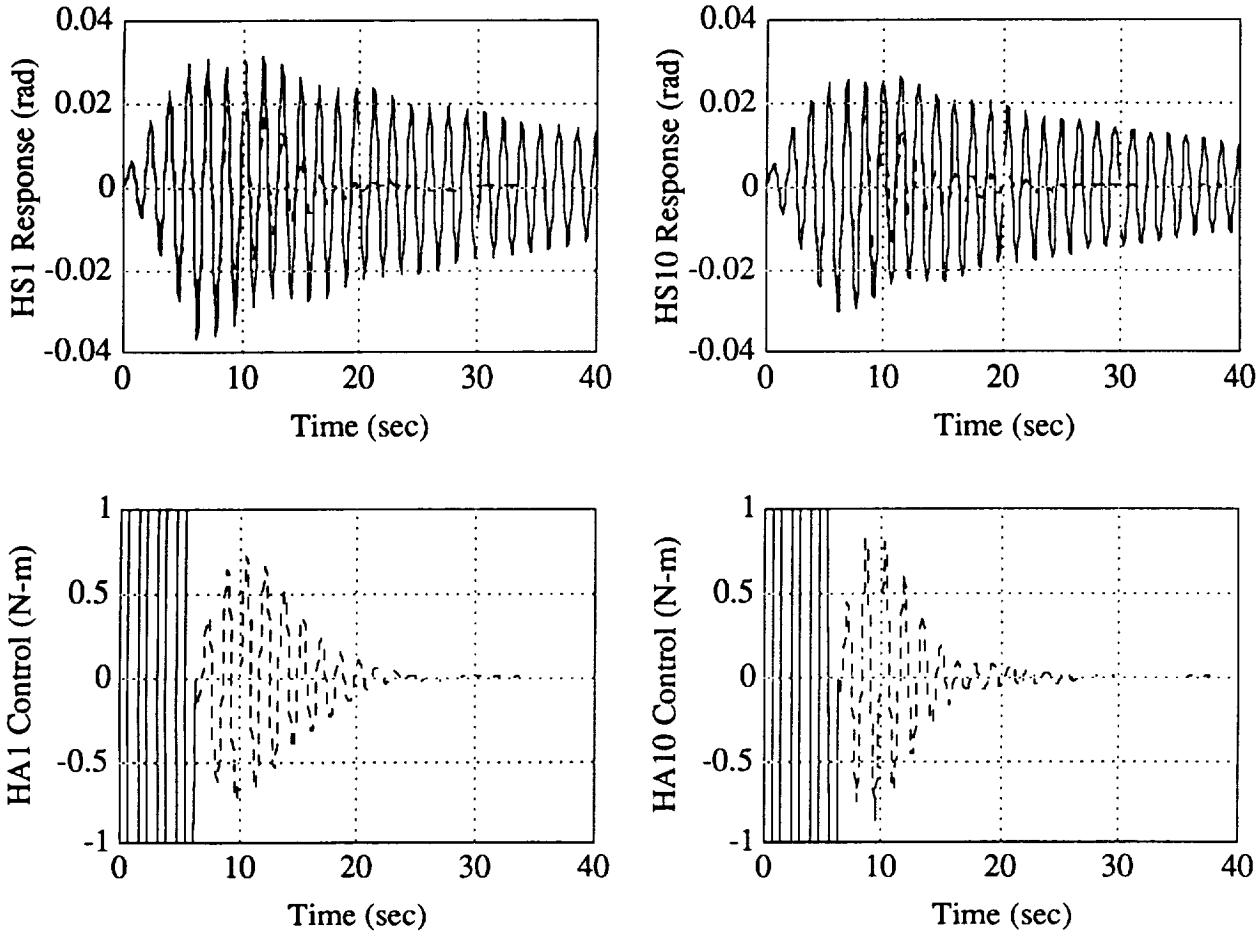

Figure 8: Open-Loop (solid curve) versus Closed-Loop (dashed) Responses